ITTO

INTERNATIONAL TROPICAL TIMBER ORGANIZATION

Project PD 162/91 - Papua New Guinea

Intensification of growth and yield studies in previously logged forest

New directions for natural forest growth and yield studies in Papua New Guinea

Denis Alder
Consultant in Forest Biometry

File: C:\Forestry\PNG\Reports\Mar'99 ITTO report.doc
Revised: 07-Apr-1999

Executive Summary

The consultant undertook three basic activities during the visit to the project described in this report. The first and major activity, requiring some two weeks work, was the preparation of a project proposal entitled *Fire, logging and site interactions in the management of lowland tropical forests*, according to the standards in the ITTO *Manual of Project Formulation*, November 1998 revision. This proposal is attached as a separate document in draft form. It will be formally submitted to ITTO by the PNG Forest Authority.

The proposal's costing and time scheduling are based around an integrated spreadsheet model that is described in some detail in the report. This should facilitate modification or adaptation of the model to use for other project proposals.

The second activity, requiring some 10 days work, was the conversion of PSP data from a FoxPro to Access database format. At the same time, 50 plots established by the PNG Forest Research Institute independently to the ITTO project were converted and merged with the main database. Only the first measurement is so far available for these plots. The new database contains now 122 plots, 260 re-measurements, and 80,722 tree records. The Access system, called PERSYST2 was not fully completed to an operational standard and requires further work. However, data entry forms of various types were tested, and no problems of operational capacity were found. PERSYST2 comprises a single file containing all data and database objects (forms, reports, etc.) of 10 MB size. For ease of backup and transfer of such data, the consultant recommends specifying the fitting of an internal ZIP 100 MB exchangeable disk drives to all new computer purchases. With the new project, PERSYST2 is likely to grow to 20-30 MB after 5 years.

It was noted that the new project will require changes to PSP measurement standards, dropping height and crown measurements, placing more emphasis on accurate diameter measurement and specifies identification, and introducing topographic survey, GPS locations, and quadrat-level site classification as a standard. These changes will impact upon database design. PERSYST2 therefore needs to be completed in co-ordination with the start-up phase of the new project.

The final activity undertaken was a 2-day training course in Microsoft Access, held at the PNG Forest Research Institute on 25-26 March for 11 participants. This course concentrated on mainstream forestry examples which would not be experienced under a conventional commercial course. Forest inventory data was used, with a typical species list and sample plot data organisation. Participants were taught the elements of database design, indexing, relationships, SQL queries, and one-to-many forms design.

Contents

List of abbreviations

Access	*Microsoft Access* 97. Registered trademark of Microsoft Corporation.
FIMS	Forest Inventory Mapping System
FRI	Forest Research Institute
GIS	Geographical Information System
GPS	Global Positioning System
ITTO	International Tropical Timber Organisation
MPF	ITTO Manual for Project Formulation
PERSYST	Permanent sample plot system
PNG	Papua New Guinea
PNGFA	PNG Forest Authority
PNGFRI	PNG Forest Research Institute
PNGRIS	PNG Resource Information System
PSP	Permanent sample plot.
SFM	Sustainable Forest Management (branch of PNGFRI)
SQL	Structured Query Language
UTM	Universal Transverse Mercator.
VBA	Visual Basic for Applications. *Visual Basic* is a registered trademark of Microsoft Corporation.

Introduction

Terms of reference

This report covers a consultancy visit to the ITTO project PD 162/91 *Intensification of growth and yield studies in previously logged forest* made from 28th February to 28th March. The terms of reference for the assignment, as amended by correspondence, were:

- To assist the PNG Forest Authority in preparing an ITTO project proposal that would take forward growth and yield studies in the lowland tropical forests to a new stage.
- To upgrade and convert the PSP database system from its FoxPro 2.5 format to an Access database that would be more easily maintained in future.
- To integrate sample plots established by the Sustainable Forest Management unit of the PNG Forest Research Institute (SFM plots) with those established by the ITTO project into a single database.
- To provide training and documentation for the new system.

In the event the first item consumed much of the time for the assignment. It was considered to be a critical activity if there was to be a smooth transition to a new phase of growth and yield studies in PNG, and was therefore given priority.

Scope of report

This report documents technical aspects of the work undertaken during the assignment. The proposal for a new ITTO project is submitted in draft form as a separate attachment. The detailed cost analysis and work plan in the proposal are based on a spreadsheet model whose use and structure are documented in the current report. This model can be used to facilitate the preparation of other proposals in a similar format, or to amend the details of the current proposal.

The structure and systems analysis of the new database, PERSYST2, are described in detail, together with a review of the information from the SFM plots in comparison to the ITTO plots.

The new project will require some amendment of PSP procedures, which has an impact on the design of PERSYST2. These proposed changes to field work are detailed and justified.

The report also contains background and reference material for the training course in MS Access given at the Forest Research Institute on 25-26 March 1999.

Origins

A note describing a project idea entitled *Fire logging and site interactions in the management of lowland tropical forests* was submitted to ITTO by the PNG Forest Authority in March 1998 and encouragement was given in November 1998 to prepare a full proposal based on this idea.

The project concept was driven by three fundamental priorities:

- The need for continuing financial support in order to maintain the existing PSPs for a further 5-year period.
- The need to deepen the investigation of some critical aspects of natural forest growth and yield. These, as the project title indicates, include forest recovery after fire, the quantification of logging damage, and development of an effective method for site classification.
- The need to have a direct application of project results in forest planning, through a concurrent integration of growth models with other PNGFA planning systems.

Current status

The complete project proposal is attached to this report as a separate document. At this level it is officially a draft, as it has to be approved by the PNGFA Board of Directors for submission to ITTO. However the attached document is complete and has been subject to internal review and revision by the PNGFA. It follows exactly the format described in the November 1998 revision of the ITTO *Manual for Project Formulation*.

The spreadsheet model of project costs

The main purpose of the present chapter is to document the spreadsheet model that underlies the proposal. As conceived in the *Manual for Project Formulation* (MPF), there is a strong logical relationship between project objectives, outputs, and activities. These are conceived as hierarchical elements, with each objective having a number of outputs, and each output a number of activities.

Costs are associated with each activity, and are also subdivided by category, which is an independent accounting classification according to an ITTO system described in the MPF.

A work plan detailing the start date and duration of each activity is a required element of the proposal. This is logically linked to the fixed and variable costs of each activity.

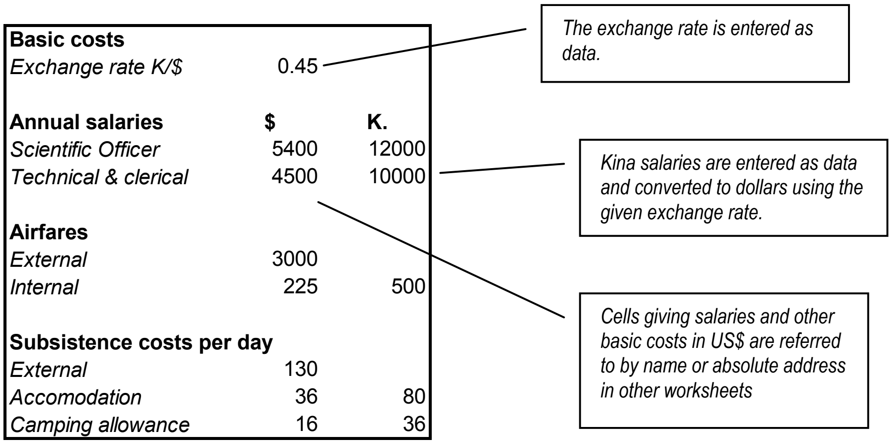
It follows that setting up the work plan, schedule of unit costs, and required tables of costs by activities and account categories by project years is a complex inter-linked task. The spreadsheet model is designed to automate this process to some degree so that it is relatively easy to adjust and refine the proposal without the need to continually edit and amend these key tables.

The model is contained in an Excel'97 file called FLSCOSTS.XLS. A copy is contained on the diskette accompanying this report. Additionally copies can be obtained from the author by e-mail at *d-alder@eurobell.co.uk*.

The various worksheets of this model will be described in the following sections.

The Basic costs worksheet

The worksheet labelled *Basic costs* is illustrated with annotations below. This sheet contains cost elements that are used repeatedly in calculating unit costs. However cost items which occur only in isolated contexts (such as the cost of a specific piece of equipment) do not need to be included in this table.



Basic costs		
Exchange rate K/$	0.45	
Annual salaries	**$**	**K.**
Scientific Officer	5400	12000
Technical & clerical	4500	10000
Airfares		
External	3000	
Internal	225	500
Subsistence costs per day		
External	130	
Accomodation	36	80
Camping allowance	16	36

The exchange rate is entered as data.

Kina salaries are entered as data and converted to dollars using the given exchange rate.

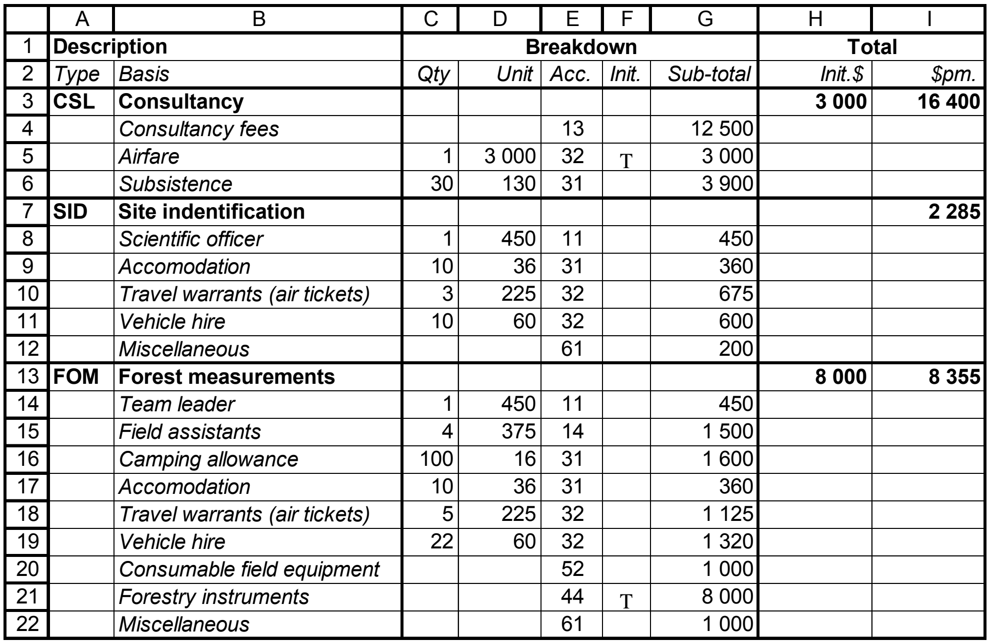
Cells giving salaries and other basic costs in US$ are referred to by name or absolute address in other worksheets

The exchange rate and Kina (local currency) figures are all entered as data. The US$ figures are calculated from the Kina values or may be entered directly. All the US$ basic cost cells are named so that they may be easily referred to on other worksheets.

Additional basic costs could be added to this table as required. The items added can be either entered directly in US$, if they are external costs such as overseas airfares and subsistence, or calculated from a local cost and the exchange rate. The items added should be given names for ease of reference on the *Unit Costs* sheet.

The Unit costs worksheet

The figure below shows part of the worksheet called *Unit costs.* It comprises a series of cost types, indicated by a three-letter code in column A. The cost type is analysed by ITTO account categories given in column E. The quantity and unit elements are an aid to calculation of the sub-total in column G. However, they do not need to be used if inappropriate, and a figure can be entered directly in G.



	A	B	C	D	E	F	G	H	I
1	**Description**		**Breakdown**					**Total**	
2	*Type*	*Basis*	*Qty*	*Unit*	*Acc.*	*Init.*	*Sub-total*	*Init.$*	*$pm.*
3	**CSL**	**Consultancy**						**3 000**	**16 400**
4		*Consultancy fees*			13		12 500		
5		*Airfare*	1	3 000	32	T	3 000		
6		*Subsistence*	30	130	31		3 900		
7	**SID**	**Site indentification**							**2 285**
8		*Scientific officer*	1	450	11		450		
9		*Accomodation*	10	36	31		360		
10		*Travel warrants (air tickets)*	3	225	32		675		
11		*Vehicle hire*	10	60	32		600		
12		*Miscellaneous*			61		200		
13	**FOM**	**Forest measurements**						**8 000**	**8 355**
14		*Team leader*	1	450	11		450		
15		*Field assistants*	4	375	14		1 500		
16		*Camping allowance*	100	16	31		1 600		
17		*Accomodation*	10	36	31		360		
18		*Travel warrants (air tickets)*	5	225	32		1 125		
19		*Vehicle hire*	22	60	32		1 320		
20		*Consumable field equipment*			52		1 000		
21		*Forestry instruments*			44	T	8 000		
22		*Miscellaneous*			61		1 000		

The cost may be constructed by reference to cells in the *Basic costs* sheet, or entered directly, as for example the *Forestry instruments* entry on line 21.

Each cost item in column G is assumed to be a per month figure unless it is classified as a fixed or initial cost by ticking column F. Ticking is done by entering a capital T. This column uses the *WP Iconic Symbols A* font, which represent T as a T.

The summary for each cost type is given on the same line as the code and is based on formulae entered in the cells. The summary is divided into a fixed and variable cost component. The fixed cost is applied only at the start of an activity. The variable cost is multiplied by the activity duration in months.

The formulae used simply sum the subtotals in column G for either fixed or variable components. For example, for the Field measurements (FOM) cost type, the fixed cost comprises the instrument purchases, with simply the formula:

=G21

in cell H13. The variable costs are calculated from the formula:

```
=SUM(G14:G20,G22)
```

in cell I13.

The illustration above is only a partial listing of the table in the project proposal, which is shown completely in Table 5 (page 20) of the attached document.

This table can be edited by adding or deleting rows from activity types, or adding completely new activity types. The main rules to observe are:

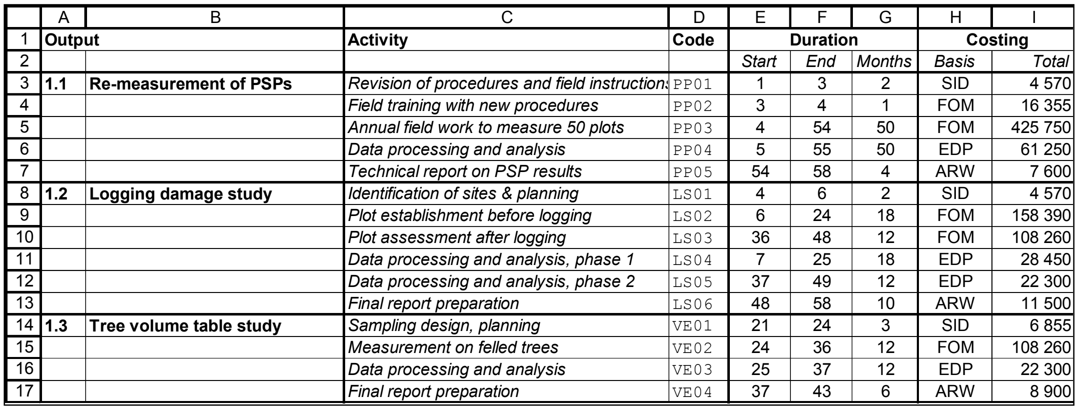
- There must be a 3-letter code in column A which is unique, and does not conflict with any other code in the list.
- The ITTO account heading must be entered in column E for each element of the cost analysis.
- Fixed or initial costs must be ticked with a T in column F. These will not be calculated on a per-month basis but once only for each activity.
- The cost elements can be either entered directly in column G or calculated from rates and quantities in columns C and D. The method is not important.
- The summary is constructed by formulae from the fixed and variable costs and must be on the same line as the code in column A.

The Activities worksheet

The list of project activities is shown in Table 2 on page 12 of the attached proposal. The activities are grouped by outputs which are themselves numbered according to the specific objectives of the project.

This table is derived from the worksheet called *Activities*. A part of this is illustrated below with row and column headings.

All the columns in this table except for column G (the activity duration) and I (the total cost of the activity) are entered as data. Columns A-C are simply text. Column D is an activity code which is used elsewhere in the model, and should have a two-letter prefix that is consistent within each output group, and a sequential activity number of 2-digits. The start and end dates, in months from the project start, are entered in columns E and F.



	A	B	C	D	E	F	G	H	I
1	**Output**		**Activity**	**Code**	**Duration**			**Costing**	
2					*Start*	*End*	*Months*	*Basis*	*Total*
3	**1.1**	**Re-measurement of PSPs**	*Revision of procedures and field instructions*	PP01	1	3	2	SID	4 570
4			*Field training with new procedures*	PP02	3	4	1	FOM	16 355
5			*Annual field work to measure 50 plots*	PP03	4	54	50	FOM	425 750
6			*Data processing and analysis*	PP04	5	55	50	EDP	61 250
7			*Technical report on PSP results*	PP05	54	58	4	ARW	7 600
8	**1.2**	**Logging damage study**	*Identification of sites & planning*	LS01	4	6	2	SID	4 570
9			*Plot establishment before logging*	LS02	6	24	18	FOM	158 390
10			*Plot assessment after logging*	LS03	36	48	12	FOM	108 260
11			*Data processing and analysis, phase 1*	LS04	7	25	18	EDP	28 450
12			*Data processing and analysis, phase 2*	LS05	37	49	12	EDP	22 300
13			*Final report preparation*	LS06	48	58	10	ARW	11 500
14	**1.3**	**Tree volume table study**	*Sampling design, planning*	VE01	21	24	3	SID	6 855
15			*Measurement on felled trees*	VE02	24	36	12	FOM	108 260
16			*Data processing and analysis*	VE03	25	37	12	EDP	22 300
17			*Final report preparation*	VE04	37	43	6	ARW	8 900

Column G contains the formula:

=INT(F3-E3)

which gives the duration of the activity to the nearest whole month.

Column I calculates the total activity cost using the simple relation:

Activity cost = Fixed cost + duration x cost per month

This is expressed in column I by the formula:

=VLOOKUP(H3,UnitCosts,8,FALSE)+VLOOKUP(H3,UnitCosts,9,FALSE)*G3

The VlookUp function locates the code in column H3 in the UnitCosts table, and returns the corresponding fixed cost in the eighth column, and variable cost in the ninth column.

The BarChart graphic

The bar chart gives a Gantt type representation of the start and end dates of activities. It is directly linked to the *Activities* worksheet, so that changes in the start and end date dates of an activity will be reflected in the bar chart.

It is also possible on the bar chart to drag the start and end points of activities with the mouse. This will be reflected in the *Activities* table and the cost calculations as start and end dates and durations change.

However, Excel does not provide for proper Gantt or other project planning charts, and there are some imperfections in the process if non-integral values are used for months. The bars on the chart may overlap slightly with the activity start symbol. It is suggested that if the mouse is used to adjust the scheduling chart directly, then the

Activities table should be edited manually to ensure integral start and end times once an approximate work plan has been achieved.

The bar chart will be found in the project proposal as Figure 1 on page 17.

The Cost DB worksheet and macro

The *Cost DB* worksheet is generated automatically from the *Activities* and *Unit Costs* table by a macro called *MakeCostsDB*. When any of the cost or activities details are changed in the worksheets, then the button marked *Rebuild List* which is near the top of the worksheet should be clicked to update the list. They will then be automatically reflected in the summary tables.

The data in this list should not be edited manually. It serves a technical purpose in allowing summarisation of costs by activities, account categories, and years using simple function in the final output tables.

The *MakeCostsDB* macro which generates and updates this table is written in Visual Basic and can be examined by clicking Alt-F11 to bring up the Visual Basic Editor. It simply re-formats data from the *Unit Costs* and *Activities* worksheets into a simple list of cost elements for each activity and account heading. It uses the start and end date information for each activity to summarise the costs by years.

The Costs by Account worksheet

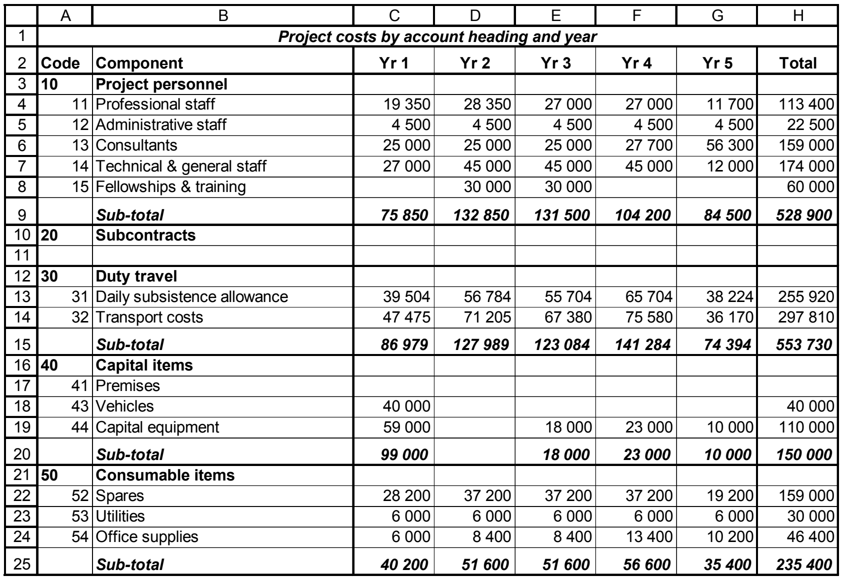
Part of the *Costs by Account* worksheet is shown below with row and column headings. The full table is shown in the project proposal as Table 7 on page 22.

This table is laid out manually with the account codes and headings from the *Manual for Project Formulation*. The entries under each year heading are summarised from the *Cost DB* worksheet using the formula:

```
=SUMIF('Cost DB'!$B$2:$B$200,$A4,'Cost DB'!D$2:D$200)
```

in cell C4, for example. This searches for the account code in A4 in the second column of the *Cost DB* table, and adds any costs from column D when the account codes are equal.

The relative and absolute addressing indicated by the $ symbols in this formula allows it to be copied elsewhere on the worksheet to give the correct year and account referencing.



	A	B	C	D	E	F	G	H
1		***Project costs by account heading and year***						
2	**Code**	**Component**	**Yr 1**	**Yr 2**	**Yr 3**	**Yr 4**	**Yr 5**	**Total**
3	**10**	**Project personnel**						
4	11	Professional staff	19 350	28 350	27 000	27 000	11 700	113 400
5	12	Administrative staff	4 500	4 500	4 500	4 500	4 500	22 500
6	13	Consultants	25 000	25 000	25 000	27 700	56 300	159 000
7	14	Technical & general staff	27 000	45 000	45 000	45 000	12 000	174 000
8	15	Fellowships & training		30 000	30 000			60 000
9		***Sub-total***	***75 850***	***132 850***	***131 500***	***104 200***	***84 500***	***528 900***
10	**20**	**Subcontracts**						
11								
12	**30**	**Duty travel**						
13	31	Daily subsistence allowance	39 504	56 784	55 704	65 704	38 224	255 920
14	32	Transport costs	47 475	71 205	67 380	75 580	36 170	297 810
15		***Sub-total***	***86 979***	***127 989***	***123 084***	***141 284***	***74 394***	***553 730***
16	**40**	**Capital items**						
17	41	Premises						
18	43	Vehicles	40 000					40 000
19	44	Capital equipment	59 000		18 000	23 000	10 000	110 000
20		***Sub-total***	***99 000***		***18 000***	***23 000***	***10 000***	***150 000***
21	**50**	**Consumable items**						
22	52	Spares	28 200	37 200	37 200	37 200	19 200	159 000
23	53	Utilities	6 000	6 000	6 000	6 000	6 000	30 000
24	54	Office supplies	6 000	8 400	8 400	13 400	10 200	46 400
25		***Sub-total***	***40 200***	***51 600***	***51 600***	***56 600***	***35 400***	***235 400***

The sub-total rows are constructed as simple sums of the cells in each section above.

The Costs by Activity worksheet

The *Costs by Activity* worksheet is very similar in construction to the *Costs by Account* sheet. The cells contain a formula which summarises the costs for a given activity and corresponding year in each row and column of the table, using the basic data in the *Cost Db* sheet.

The worksheet is shown as Table 6 in the project proposal.

The Activity by Account worksheet

This worksheet, which appears as Table 8 in the proposal, summarises costs over the whole project life by main account headings and activities. The principal is similar to the preceding examples, and depends directly on the data in *Cost DB* which is summarised using the SUMIF worksheet function to match a constructed activity and account category code with the values in the *Cost DB* listing.

Conclusion

The attached project proposal for a new 5-year ITTO project is self explanatory and complete and should be referred to for details of the proposal itself. The material in this section is intended to facilitate amendment of the underlying costing and scheduling model if necessary.

Revisions to the PSP database

The need to update PERSYST

The data from the ITTO permanent sample plots is stored in a computer system called PERSYST. The data coding system and operations of this database are described in project reports by Romijn (1994a, b). The status of the data, and a summary listing of the data structures, are described in Alder *et al*. (1998). The total number of plots in this system is 72, with a total of 210 measurements.

The PSPs established by the Forest Research Institute have been entered by various methods. However, currently there are 50 plots for which data has been entered using a copy of the PERSYST database. The data on disk comprises the initial plot measurement only.

PERSYST is a system using FoxPro 2.5. It is not a compiled program, and requires the FoxPro package to be installed before it can operate. This is no longer available in this version. Currently FoxPro is being issued as version 7, and involves many changes in operation and file structures relative to version 2.5, which dates back to 1992. The PERSYST program cannot be run on current versions of FoxPro.

The consultant proposed that the database should be updated to a version using current software, and recommended for this purpose Microsoft Access, rather than FoxPro. The main advantages are wide availability as pre-installed software, ease of use and familiarity as part of the Microsoft Office suite, and good processing capabilities using SQL and Visual Basic.

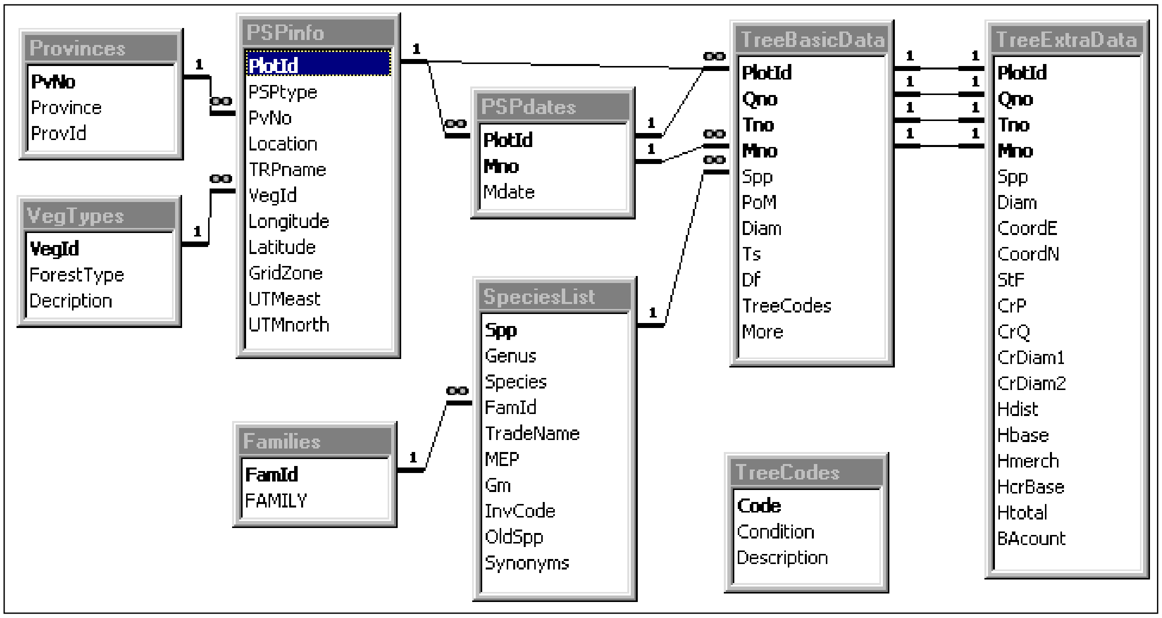
Database design

In Access, a database is stored in a single file, as compared with FoxPro, where each database object (data table, index, form, report, etc.) uses a different file. The new version of PERSYST is stored in a file called PERSYST2.

The various data tables in this file are shown in the figure below with their relationships and linkages.

Plot identity and location information is stored in the table **PSPinfo**. This information is imported from the PERSYST table PSP_LIST with minor modifications. However measurement dates are not stored in the same file, as in PERSYST, but in a separate file called **PSPdates**. This is a more flexible scheme that allows data to be extracted by date or measurement number using standard SQL statements. It also imposes no limitations on the number of measurements in the system.

Figure 1 Data tables and relationships in the PERSYST2 database



The tree data from the PERSYST files is split into two tables, called **TreeBasicData** and **TreeExtraData**. The former contains measurements and observations made at every measurement. The latter contains additional observations of tree location, crown dimensions, tree height, etc. which are made only initially and for recruits.

Each tree and measurement in the tree tables is uniquely identified by a key derived from plot identification (*PlotId*), quadrat number (*Qno*), tree number (*Tno*), and measurement number (*Mno*).

The species list table (**SpeciesList**) is directly imported from the equivalent table in PERSYST, but unused fields in the latter file have been dropped. Considerable work was also expended establishing *referential integrity* between the species list and tree data files, as indicated by the **1-**∞ symbol joining fields in Figure 1. This required that duplicate and null codes be cleaned from the species list file. In the tree data, it was necessary to check and edit all 'lost' codes, *ie.* those which had no entry in the species list.

Family names in the species list were abbreviated to the first four letters in the field *FamId*, and a separate table of family names established (**Families**).

The tree condition codes (*TreeCodes*) observed for each tree and measurement are linked to the codes and descriptions in the **TreeCodes** table. This linkage is not a formal one within Access, and is therefore not shown on the figure, but will be managed by a small Visual Basic function. This is necessary because the tree codes are stored in a comma separated list, and there may be several for each tree.

Referential integrity

The relational linkages shown in the diagram by the **1-**∞ symbols indicate that referential integrity has been established and checked on a one-to-many basis. The **1-1** symbol joining the *TreeBasicData* and *TreeExtraData* tables indicates a one-to-one relationship.

These concepts do not exist in FoxPro, but are essential to check the logical validity of the database, to verify data entry, and to accelerate operations that draw on data from more than one table. Equivalent processes which are handled automatically and efficiently in Access need to be programmed in FoxPro. For example, if a species code is changed in the species list due to a change in botanical nomenclature, Access will do a *cascade update* operation to automatically correct all species codes in the database to reflect the change.

Establishing referential integrity with the existing data has taken some time, and many corrections have had to be made throughout the database to complete this process.

Conversion of PERSYST data

A Visual Basic program was written called *AutoAdd* to automatically append the FoxPro tree data to the Access database. This splits the original files into the basic and extra measurements. It is designed to work as an updating routine, so it does not duplicate records already in the system, but overwrites them. This will allow data entry and editing to continue on PERSYST. The data can be added to PERSYST2 at any time without any loss or duplication of information.

Data entry forms

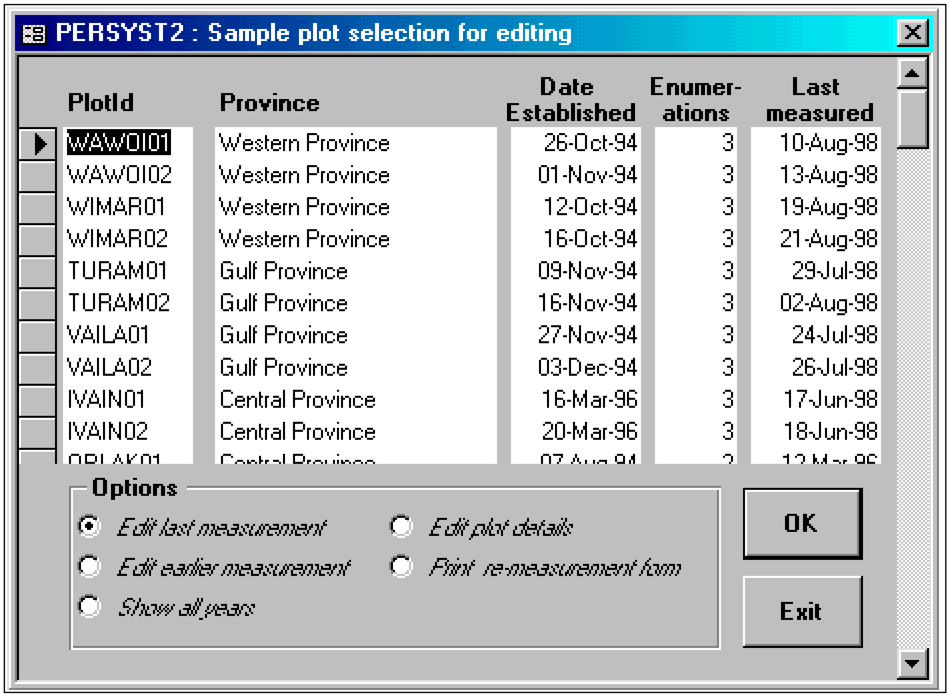
The concepts for the introductory screen and data entry forms have been completed. Some modifications are needed in relation to changed measurement procedures, as will be discussed. Time has also not permitted the forms to be tested in use.

The initial data entry form is a plot selection screen which also has several options, as shown in Figure 2. A plot to be edited is selected. It is then possible to edit the last (most recent) measurement, an earlier measurement, or data for all measurements combined. These options are operational.

Figure 3 shows the tree data editing screen with the extra measurements box displayed. The latter appears when the cursor is moved to the 'More' field, but will normally be invisible.

Several aspects of these forms need to be optimized, and there will be minor changes in layout in future. It is planned, for example, to combine the *Sf* and *Df* status fields with the tree codes.

Figure 2 Plot selection screen for data entry with PERSYST2



PERSYST2 : Sample plot selection for editing

PlotId	Province	Date Established	Enumer-ations	Last measured
WAWOI01	Western Province	26-Oct-94	3	10-Aug-98
WAWOI02	Western Province	01-Nov-94	3	13-Aug-98
WIMAR01	Western Province	12-Oct-94	3	19-Aug-98
WIMAR02	Western Province	16-Oct-94	3	21-Aug-98
TURAM01	Gulf Province	09-Nov-94	3	29-Jul-98
TURAM02	Gulf Province	16-Nov-94	3	02-Aug-98
VAILA01	Gulf Province	27-Nov-94	3	24-Jul-98
VAILA02	Gulf Province	03-Dec-94	3	26-Jul-98
IVAIN01	Central Province	16-Mar-96	3	17-Jun-98
IVAIN02	Central Province	20-Mar-96	3	18-Jun-98
ORLAK01	Central Province	07-Aug-94	2	13-Mar-96

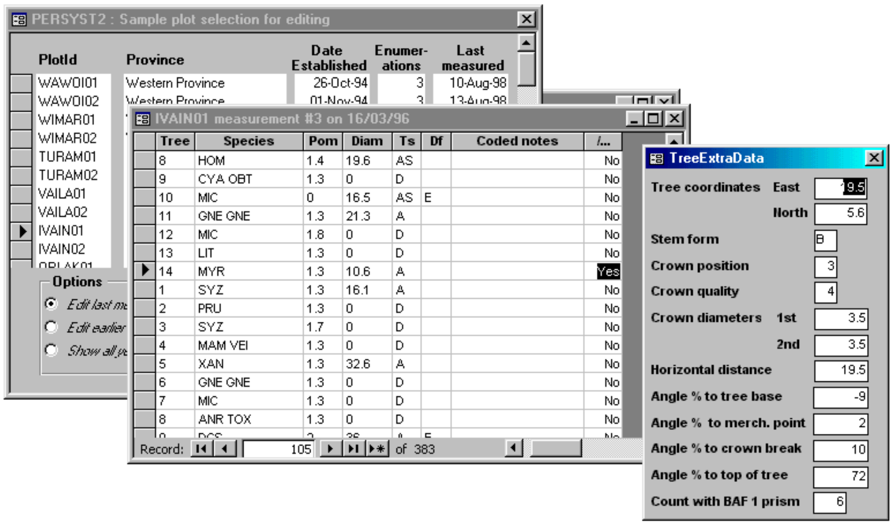
Options

● *Edit last measurement* ○ *Edit plot details*

○ *Edit earlier measurement* ○ *Print re-measurement form*

○ *Show all years*

However the existing system serves to test the concepts of the database. All the tree data is in a single table, which currently contains 80,722 records. When a plot is selected (via the screen in Figure 2), then the records for the required plot and year need to be extracted via an SQL query and presented on the form in Figure 3. This process is effectively instantaneous, confirming that there is no slowing down of the system due to the large size of the database.



PERSYST2 : Sample plot selection for editing

PlotId	Province	Date Established	Enumer-ations	Last measured
WAWOI01	Western Province	26-Oct-94	3	10-Aug-98
WAWOI02	Western Province	01-Nov-94	3	13-Aug-98
WIMAR01				
WIMAR02				
TURAM01				
TURAM02				
VAILA01				
VAILA02				
IVAIN01				
IVAIN02				

Options
● Edit last me
○ Edit earlier
○ Show all ye

IVAIN01 measurement #3 on 16/03/96

Tree	Species	Pom	Diam	Ts	Df	Coded notes	/...
8	HOM	1.4	19.6	AS			No
9	CYA OBT	1.3	0	D			No
10	MIC	0	16.5	AS	E		No
11	GNE GNE	1.3	21.3	A			No
12	MIC	1.8	0	D			No
13	LIT	1.3	0	D			No
14	MYR	1.3	10.6	A			Yes
1	SYZ	1.3	16.1	A			No
2	PRU	1.3	0	D			No
3	SYZ	1.7	0	D			No
4	MAM VEI	1.3	0	D			No
5	XAN	1.3	32.6	A			No
6	GNE GNE	1.3	0	D			No
7	MIC	1.3	0	D			No
8	ANR TOX	1.3	0	D			No

Record: 105 of 383

TreeExtraData

Tree coordinates	East	19.5
	North	5.6
Stem form		B
Crown position		3
Crown quality		4
Crown diameters	1st	3.5
	2nd	3.5
Horizontal distance		19.5
Angle % to tree base		-9
Angle % to merch. point		2
Angle % to crown break		10
Angle % to top of tree		72
Count with BAF 1 prism		6

Figure 3 Tree data entry screen with extra data form visible

Other modules required

Before PERSYST2 can become an operational system, the following basic facilities need to be added:

□ Editing of plot identification information

□ An entry format and data table for quadrat-level site codes.

□ Plot re-measurement form

□ Data checking and error correction

□ Progress summary reports

None of these is a major task, but it has not been possible within the constraints of the present assignment to complete them. Because the existing PERSYST system provides these facilities, and some procedural changes are needed to field measurements that will impact on forms and database design, it is not recommended to undertake this work until a new PSP project is in place.

Revisions to re-measurement procedures

Some modifications to PSP procedures are needed in future. The consultant has discussed these in various presentations, and has also recommended that they should not be introduced until a new PSP project starts up. Ongoing plot enumeration under the existing project should continue to use established procedures and forms.

The modifications required may be summarised as follows:

□ The routine measurement of height, crown diameter, and competing basal area should be discontinued. These are all slow and inaccurate measurements which contribute little useful information and are excessively costly to gather.

- The system of coded notes should be revised to establish more hierarchical mnemonics. Some codes which are entered in separate fields (the *Sf* and *Df* codes) should be amalgamated with the more general codes). Some important codes (for fire damage, for example) have not been defined.

- Tree crown position (competitive status) should be assessed routinely at each measurement. Although it does not change very much over time, the consecutive assessments will serve to verify this subjective measure. It also has a useful correlation with increment.

- Tree form and canopy vigour should be dealt with as abnormal or exception conditions within the coded note system, rather than by a simple score. Only trees of abnormal crown condition or form need a note.

- Tree co-ordinates should be retained and measured for each recruit tree. A checking procedure needs to be introduced in the field using plot maps to identify errors, as the existing large mass of co-ordinate data has never been field checked. Slope corrections need to be properly used when measuring co-ordinates, as many trees occur on steep embankments.

- Diameter measurement for large and malformed trees needs to be done more rigorously. Aluminium 2-stage ladders need to be carried in the field allowing direct measurement up to 5 m above ground. For other trees which are malformed (eg. strangled trees) or have buttresses or fluting more than 5 m high, then Relascopes should be used to obtain an objective measurement. At the moment, the diameter of such trees are simply guessed ('estimated').

- There is no provision within the existing procedures, forms or database for a change in the point of measurement, other than by abandoning one POM and starting a new one. This needs to be amended.

- Species codes are carried forward on the re-measurement form and often accepted uncritically. In some cases, species coding has become mixed up to due to indexing errors within FoxPro. In other cases, wrong identifications have been made or wrong codes used. The existing form provides no space to write a species name. Too many species are field identified only to the generic level. There needs to be a general tightening of rigour and improvement of procedure with regard to species identification.

- Palms need to be included within the plot on the same basis as trees. At the moment they are ignored, giving a quite false impression of the stand density of some plots.

- A simple topographic survey of all plots needs to be recorded, using a level along quadrat boundary lines. Many plots have extreme topography, and their site characteristics cannot be properly understood without this information.

- GPS co-ordinates are needed for all plots. Currently 45 PSPs have co-ordinates recorded from a total of 122 (72 ITTO and 50 SFM plots). These also need to be checked against maps and repeated on more than one occasion for verification. Co-ordinates are needed in longitude and latitude as well as UTM measure.

- A site evaluation system based on coded notes is required for each quadrat at each measurement. The procedures for the ITTO plots do have a system of site scoring at the plot level, but this has never been applied. A plot level assessment is also too generalised on a 100 x 100 m unit.

Introducing these measures requires a re-designed field form, a field training exercise on one plot, and periodic monitoring of re-measurements to ensure standards are maintained. The database also needs to be modified in minor respects in co-ordination with the field form.

These changes cannot be introduced within the terminal phase of the present project, but should form one of the earliest activities in the new project. This is provided for in the project proposal attached.

Computer hardware recommendations

Unlike the FoxPro based PERSYST system, PERSYST2 places the entire database within a single file. This has many advantages in terms of simplicity of installation, management and data back up, and ensures that the database is always internally consistent. With the FoxPro system, index files can be copied which are out of synchronisation with their associated data files, causing severe data corruption. As noted, some plots have muddled species codes probably caused by this, and will require extensive editing against original forms.

However, a disadvantage is that the single database file (PERYST2.MDB) can be become quite large. Currently with 80,000 trees and 260 plot re-measurements, the database is about 10 MB.

With the new project this may be expected to double to about 20 MB over a 4-5 year period.

The consultant considers that in this environment, conventional floppy disks are obsolete as a means of data storage and backup, and ZIP disks should be used instead. ZIP disks store 100 MB, and cost about $150 as an internally fitted unit.   They should be ordered as a standard with all new computers, and retrofitted to existing systems.

Conclusion It has not been possible within this assignment to fully develop a revised version of PERSYST. However, several useful steps have been accomplished:

- The groundwork for the new system has been laid in terms of database design.

- All the existing data has been converted (from both ITTO and SFM) plots and consolidated within the Access database PERSYST2.

- A data conversion program has been written that can be used to update PERSYST2 for new data added to the old PERSYST files during the remaining life of the current project.

- The feasibility of a large Access PSP database has been tested by setting up a data entry and editing system with various options using the full data set. Performance has been found to be excellent, in spite of the large amount of data involved.

- The procedural changes that ill be needed in future for the next phase of PSP work have been identified and defined.

Training programme in Access

Programme outline

The Access training programme was given on the 25th and 26th March to 11 participants from various sections of the Forest Research Institute and the Forest Authority. Five computers were used for the practical work, with a sixth used for demonstrations with a multimedia projector.

The workshop started with the importation of inventory data, a species list, and a list of inventory projects from FIPS FoxPro files into Access. The resultant data tables were modified and relationships set up between them. A graded series of queries were then demonstrated using one or several tables, as lists and with summaries and calculations.

The method of building one-to-many forms was demonstrated using the plot header and tree data situation that is common in forestry. The use of the report generator was explained using a species list grouped by commercial classes as an example.

Finally, more complex queries for cross-tabulating data were explained, with the species by diameter class stand table being used as an example.

The methods available for copying query and report results to Word were shown, as was the simple import and export of data to Excel.

The sections which follow give reference material and a more detailed outline of the various activities undertaken by the participants.

Importation and conversion of data

On each computer in the workshop , a working directory called C:\TRAINING was set up containing some data files copied from the PNGFA's FoxPro-based forest inventory system, FIPS. These files were as follows:

F_CODE.DBF	*FIPS species list*
F_MASTER.DBF	*List of inventory projects*
FD010101.DBF	*Data files from three projects (one project has three inventory blocks, the other two have one each).*
FD010201.DBF	
FD010202.DBF	
FD010203.DBF	
FD010301.DBF	

It was explained to the participants how to open a blank, new Access database. They were asked to import the files F_CODE and

F_MASTER using the Access menu selections **File -> Get external data -> Import.**

It was explained that for a modern system, obscure names such as F_CODE or F_MASTER were no longer appropriate. It was shown how the imported data tables could be renamed via the **Edit -> Rename** menu. They were renamed to, respectively, *SpeciesList* and *Projects.*

It was noted that names of data tables could be of any length and include various special characters. However, it would be found more practicable at a later stage to keep names fairly concise and avoid using embedded blanks[1].

The structures of these tables were then altered by selecting each table and clicking the **Design** button. The use of the design screen was explained. Field names should be constructed to be concise and meaningful. If a data item was intrinsically narrow (*eg.* a 2-digit number), then the field name should be kept short also. Otherwise wide columns would be needed to display a long field name, restricting the space available on the screen to show other data.

The significance of text, date and numeric fields was explained. Numeric types could be integer, long integer, single or double. The meaning of these terms was explained. The Help system was demonstrated for reference to such information and the related formatting options.

During these exercises, the participants used both the *Design* and *Datasheet* views of tables, and became familiar with the key strokes and methods for switching between them.

This section provided an exercise in the following areas:

□ Simple importation of data tables from FoxPro or other DBF formats.

□ Renaming of data tables, using meaningful and appropriate rules.

□ Changing the structure of data tables, renaming fields and changing their data type and format.

[1] Embedded blanks require the use of […] around the name in SQL statements. They are also incompatible with Microsoft's Client/Server SQL system that may be used for large database applications in combination with Access.

Conversion of inventory data

One of the inventory data files was imported and shown to the participants. It consisted of a complex format with many diameter, height, tree form and species fields across each record. It was explained that this type of format represents bad design in a database system, and probably arises from the fact that FIPS may have been converted into FoxPro from an older program in a language such as FORTRAN or COBOL.

To disentangle the data into a format with one tree measurement per record requires a program in Visual Basic. A suitable program was supplied to the participants, and they were shown how to run it from the Visual Basic editing screen. They were required to manually edit the code to input the correct working directory on their computers.

The result of this was a single table called *InvData* containing all the tree data from the five inventory blocks used for the demonstration. This comprised some 48,000 tree records, which was sufficiently large that the various exercises undertaken later would be realistic in scope.

This section demonstrated to participants:

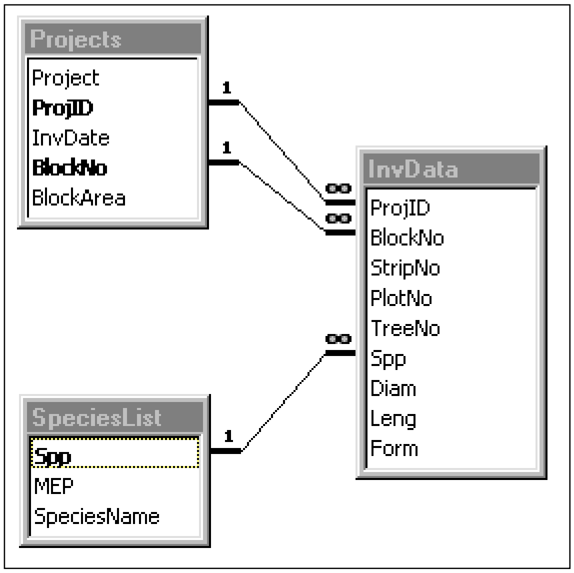
- Good and bad concepts of data record design.

- Where Visual Basic programs were found in Access and how to enter, edit and run them.

The Visual Basic program itself was not discussed in any detail, as it was outside the scope of such a short introductory course. However, for the more interested participant, it can be explored using the Help system for explanations. It provides examples of methods and procedures commonly used for sequentially processing records from data tables in Access, and shows how an external FoxPro file can be opened directly.

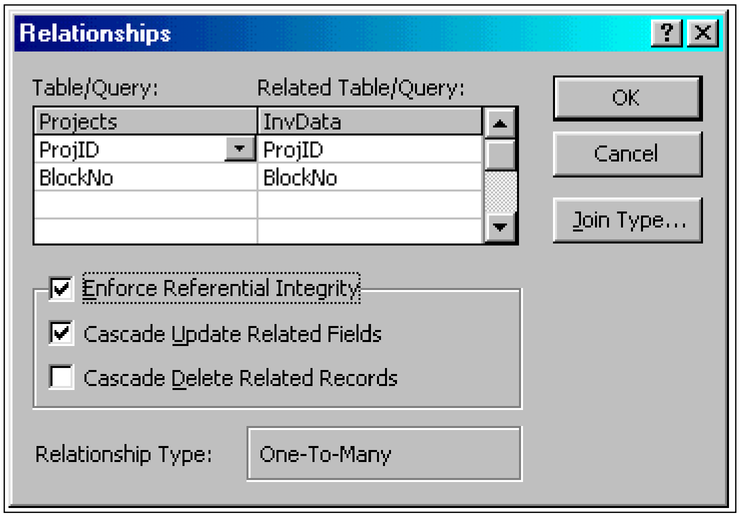
Setting up keys, indexes and relationships

The method of setting up indexes, and the types of indexes were explained. Unique indexes and primary keys were set up on the *Spp* field in the *SpeciesList* table using the **Tables -> Design** screen. A non-unique index was set up on the *ProjId* and *BlockNo* fields in the *Projects* table. The participants were shown how the latter two fields could be combined into a multiple-field primary key. Each field is selected using the **Ctrl-Click** mouse operation (standard for multiple non-adjacent selections in Windows), and then the menu selection **Edit -> Primary key** used to combine them as a primary key. In the *InvData* table, the corresponding field *Spp*, *ProjId* and *BlockNo* were selected as non-unique keys.



The participants were then shown how to activate the *Relationships* window via the **Tools -> Relationships** menu selection. The three tables in the database were added, and the primary key fields in the *SpeciesList* and Projects tables linked to the *InvData* table. The results should appear as in the figure opposite.

Various common problems were encountered whilst trying these actions. Error messages occur if the fields which have been joined in the different tables are of incompatible data types. Odd and spaghetti-like linkages arise if the dialog screen to set up relationships is not completed properly.



Table/Query:	Related Table/Query:
Projects	InvData
ProjID	ProjID
BlockNo	BlockNo

☑ Enforce Referential Integrity

☑ Cascade Update Related Fields

☐ Cascade Delete Related Records

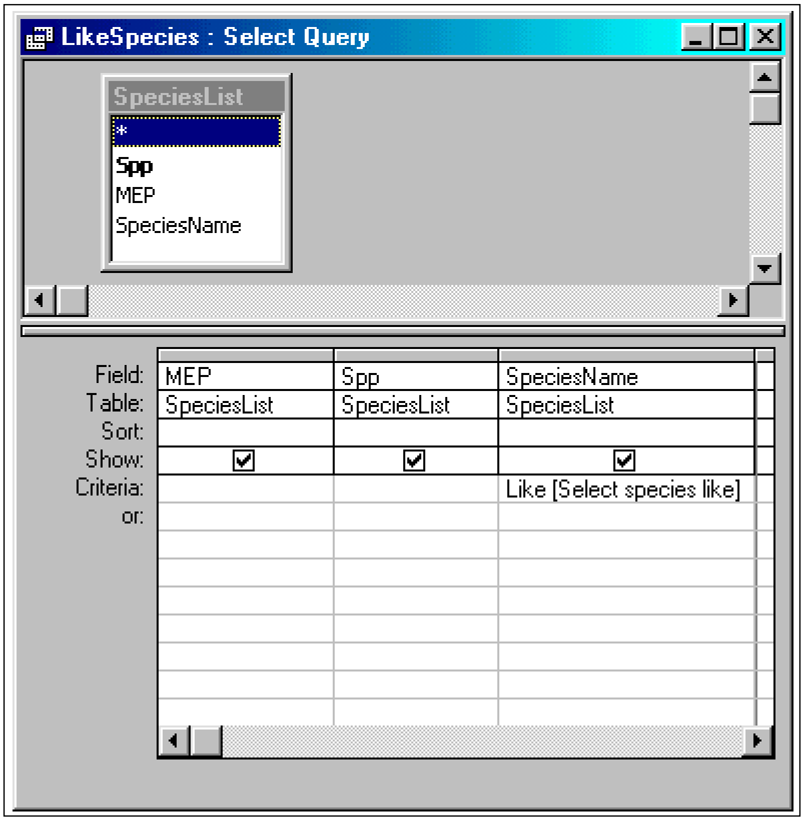
Relationship Type: One-To-Many

The correct appearance of the relationships dialog is shown to the left. The same fields names are given for both tables, and the *enforce referential integrity* box checked. The relationship type should show as *one-to-many*. If it appears as *indeterminate*, then there is an incompatibility between the linked fields. Various error messages can also arise due to the joining of fields of different data types or fields which are not indexed. For a *one-to-many* relationship, the *one* side must be a primary key.

Defining and using queries

Queries are in many ways the core of an SQL database such as Access. Whereas in FoxPro almost all operations to join data tables and extract and summarise information from them require programming, in SQL databases this is done via the establishment of relationships and the use of queries. Consequently the workshop devoted most time to this aspect.

A query on a single table was first demonstrated using the species list as an example. Various options were progressively explored, culminating in the use of the LIKE operator to select species according to flexible criteria.



Field:	MEP	Spp	SpeciesName
Table:	SpeciesList	SpeciesList	SpeciesList
Sort:			
Show:	☑	☑	☑
Criteria:			Like [Select species like]
or:			

The query design screen is shown for this first example. The species table was added to the query, and the fields *MEP*, *Spp* and *SpeciesName* selected in order.

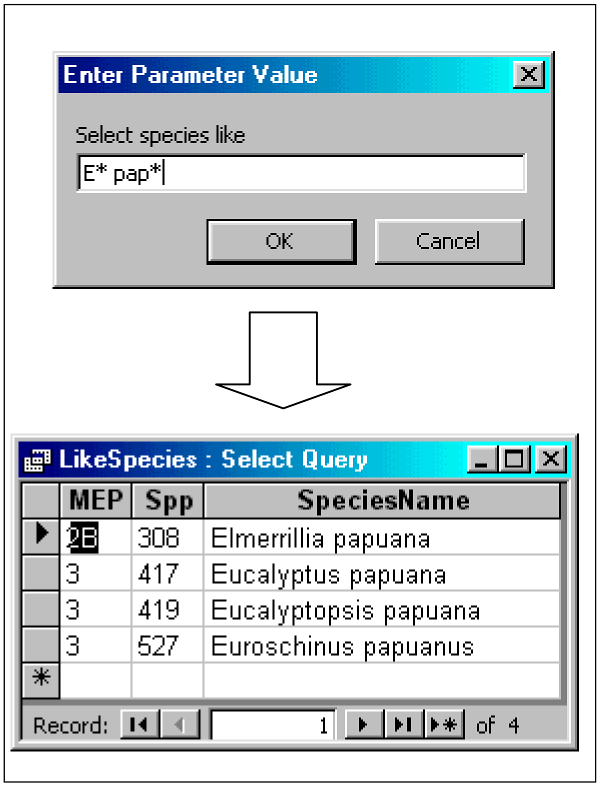
The participants experimented with these steps, and ran the query to show the full species list. They were shown how the columns could be re-sized and re-ordered using the mouse.

A criteria was then added based on the LIKE operator and a variable name which also functions as a prompt when the query is run.

The LIKE operator takes wildcards such as * (any character string) or ? (any individual character), character ranges enclosed in square brackets (eg. [a-c] allows any character a, b, or c) and literals, to form a match. The example dialog from the above query selects any species with the specific name start *'pap'*. The participants were shown how they could obtain full details on this and other operators via the Help system.

The results from this query are shown in the example below, where the selection string *E* pap** is given. This selects genera starting with *E* and species starting with *pap*.

The design screen has a corresponding text description as an SQL query. Particpants were shown how to move between the two views of a query. The SQL query for this example is:



Enter Parameter Value

Select species like

E* pap*

LikeSpecies : Select Query

MEP	Spp	SpeciesName
2B	308	Elmerrillia papuana
3	417	Eucalyptus papuana
3	419	Eucalyptopsis papuana
3	527	Euroschinus papuanus

Record: 1 of 4

```
SELECT SpeciesList.MEP,
SpeciesList.Spp,
SpeciesList.SpeciesName
FROM SpeciesList
WHERE SpeciesList.SpeciesName
Like [Select species like];
```

Participants were shown how the SQL query could be edited directly. Some operations are clearer in SQL view, but generally any operation or type of query can be designated in either view. The main advantage of familiarity with SQL, rather than the design screen, is that SQL is necessary when programming queries within Visual Basic. It is also useful when transferring to other SQL databases to be familiar with the command syntax and usage.

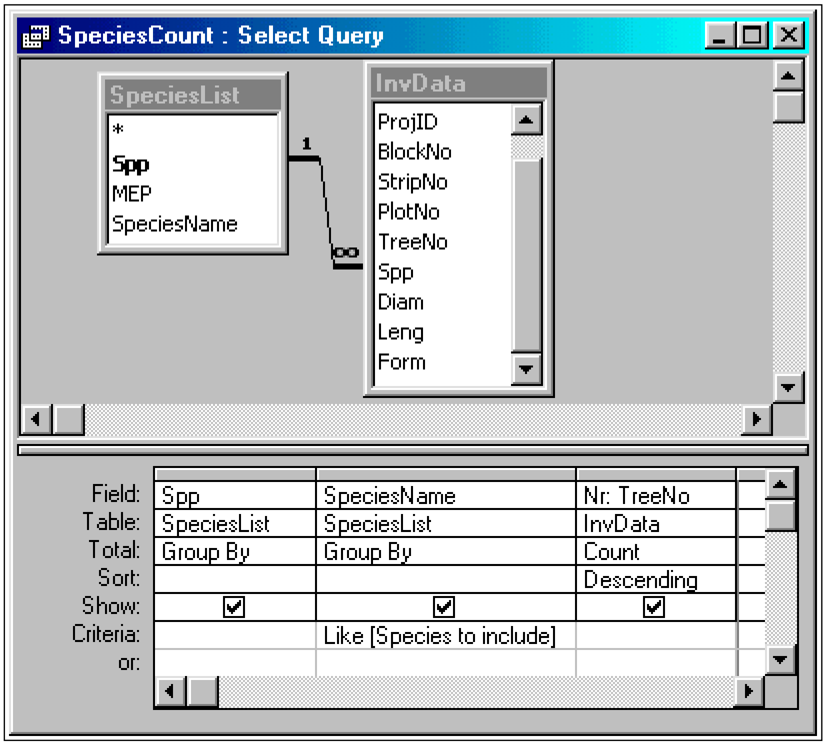
A two-table query with summarisation of data

The method for constructing a query with two tables was then demonstrated, using as an example a query which shows the numbers of trees in the inventory data grouped by species. The design screen is shown in the figure overleaf.

In this figure, the tables are added to the design screen using the **Query -> Add tables** menu selection. The link between species code *Spp* in the species list and the inventory data appears automatically because of the relationships that have been set up in the database. Relationships can also be defined manually and temporarily within queries by dragging and dropping field names between tables.

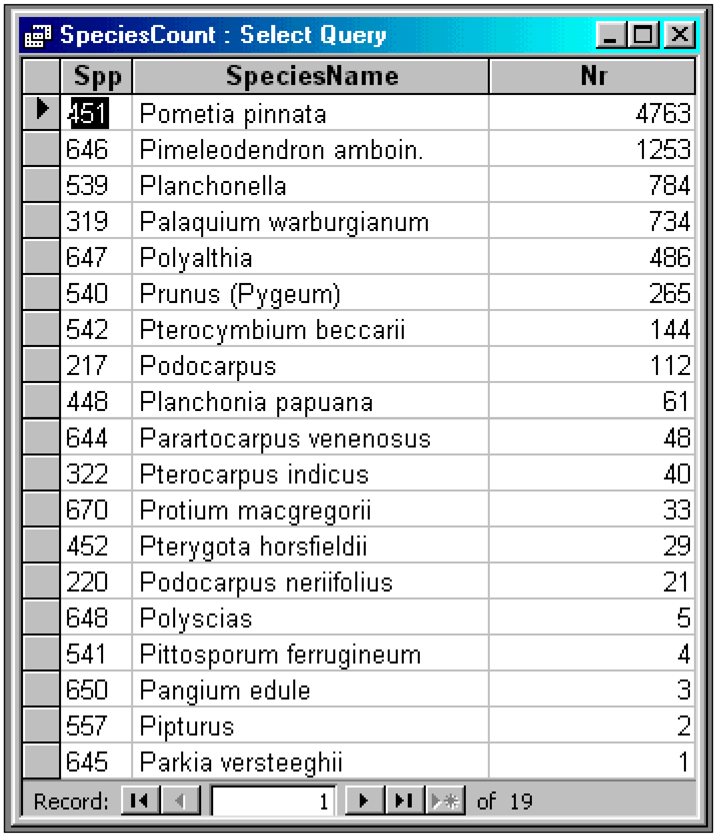
This query includes a *Total* row. This appears when the Σ icon is selected from the *Query* tool bar.

Under the *TreeNo* field, the *Count* aggregate operator has been selected. This counts all valid records within the designated groups. Preceding the *TreeNo* field name, the symbols *Nr:* have been typed manually. This is a column heading and field name in the output query, which would otherwise automatically default to *CountOfTreeNo.*



Field:	Spp	SpeciesName	Nr: TreeNo
Table:	SpeciesList	SpeciesList	InvData
Total:	Group By	Group By	Count
Sort:			Descending
Show:	☑	☑	☑
Criteria:		Like [Species to include]	
or:			

The output data is sorted in descending order by tree numbers. This is done by selecting the *Descending* option on the *Sort* row under the *TreeNo* column.



Spp	SpeciesName	Nr
451	Pometia pinnata	4763
646	Pimeleodendron amboin.	1253
539	Planchonella	784
319	Palaquium warburgianum	734
647	Polyalthia	486
540	Prunus (Pygeum)	265
542	Pterocymbium beccarii	144
217	Podocarpus	112
448	Planchonia papuana	61
644	Parartocarpus venenosus	48
322	Pterocarpus indicus	40
670	Protium macgregorii	33
452	Pterygota horsfieldii	29
220	Podocarpus neriifolius	21
648	Polyscias	5
541	Pittosporum ferrugineum	4
650	Pangium edule	3
557	Pipturus	2
645	Parkia versteeghii	1

Record: 1 of 19

This query produces the result shown to the left when the *Species to include* are given as P*:

Participants were asked to develop their own queries, based on the above single and two table examples. They found this quite difficult, and substantially different to simply working through guided examples.

Generating new tables with queries

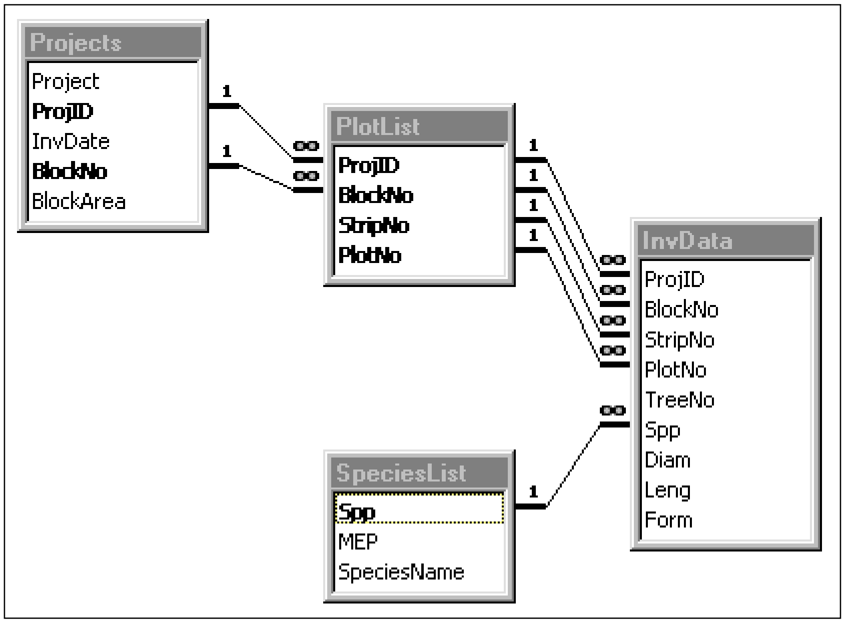
The next example covered showed how a new data table could be created by a query. The objective was to produce a plot header file to help organise the inventory data. This was done by grouping the tree data by the fields needed to identify uniquely each plot. The SQL form of the required query was:

```
SELECT ProjID, BlockNo, StripNo, PlotNo
        INTO PlotList
        FROM InvData
        GROUP BY ProjID, BlockNo, StripNo, PlotNo;
```

This differed from earlier examples by including the INTO keyword. The results of this query are saved permanently in the table *PlotList*.

Re-organising the database to group data by plots

With the list of plots generated, it was possible to re-organise the database in a more typical fashion for forestry data. Originally, the design shown on page 20 was not very logical. Tree data is linked to projects and blocks, but there is no implied organisation by plots.



The newly generated table of plot headers was indexed by each of the fields *ProjID*, *BlockNo*, *StripNo*, and *PlotNo*, which were also collectively defined as a primary key for the *PlotList* table. The relationships in the database were then re-organised to give the diagram shown. This links plots by blocks and projects, and trees by plots.

Producing a one-to-many data input form

Having set up the relationships between the plot header information in *PlotList* and the tree data in *InvData*, it is an easy matter to create a one-to-many data input form that has a header section for the plot, and a sub-section for the trees within the plot. The standard form wizard is used. The fields from the *PlotList* database and the *InvData* database are selected for the form. The Wizard automatically detects that a one-to-many form is required. By selecting all the default options from the wizard, a reasonable initial form can be produced.

The participants were shown how the layout of this initial data entry form could be improved by moving around fields and text in design mode. The text labels supplied are simply field names. These could be replaced by more appropriate labels. Different fonts can be used for clarity. The result is a form that would be quite useable for inventory data entry.

Conclusion

This summarises the main materials covered in the training course. Some other examples of queries were covered, and a brief look was given to the report generator.   Clearly, a longer course is needed to build real skills in the use of Access, but a useful introduction was given, and it would have certainly helped those who are sufficiently interested to make progress on their own.

It is notable that developing a course of this nature requires considerable preparation. The consultant used two full days to prepare and test examples for a course of two days duration, but was not able in that time to prepare very much in the way of printed hand-outs. It can be estimated that the real preparation time is probably of the order of 2:1 in terms of preparation time relative to actual teaching time.

Conclusions

New growth and yield project

This report has described the a spreadsheet model that was used to analyse costs for a new growth and yield project in Papua New Guinea. The complete project proposal in draft form is provided as an attachment to this report for reference.

This new project has to go through a number of remaining stages, including review and approval by the Board of the PNG Forest Authority, and submission for technical review to ITTO. Both of these stages may result in modifications to the attached draft. The documentation in the present report will assist in the process of making such changes.

Revision to the PSP database

Currently PSP data captured under the project is being entered and processed via the PERSYST program written to work under FoxPro 2.5. This program works very well and can continue to be used until the end of the present project and beyond.

The consultant during the present visit has begun the process of installing new software. The data files from PERSYST have all been converted into an Access database called PERSYST2. In addition, 50 PSPs established by the PNG Forest Research Institute independently of the project have been added to the database. For these plots, only the first measurement data is yet available.

Some facilities have been completed in the Access database, notably plot data entry. These provide a test of operational speed and efficiency, with a total data set of 122 plots, 260 plot re-measurements, and 80,000 tree records. Access is fully efficient with this amount of data, and this type of database can continue to be developed in future.

There are a number of advantages in using Access. Other major database software used by the PNG Forest Authority (PNGRIS, FIMS) is based on Access, minimising the problems of staff training and recruitment of technical personnel. The Access database is more closely integrated and has better internal checks than FoxPro, avoiding a number of serious problems that can arise with the latter. The FoxPro software used by PERSYST is now out of date and unavailable, so that sooner or later a major conversion exercise will in any case be required. Access is normally installed on new computers as part of the Microsoft Office suite, whereas FoxPro would need to be purchased separately at some expense for each machine on which the software is to be run.

The consultant is of the view that the further development of the Access-based PERSYST2 database to a fully operational standard is not an immediate requirement. Some revisions are needed to PSP measurement standards, and these need to be embodied in the forms and coding systems of the new database. This cannot be done until a new project is approved. It is therefore suggested that completion of PERSYST2 should be undertaken as an early activity for the new project.

PERSYST2 incorporates a routine for conversion of data from PERSYST, and there will therefore be no difficulty if the latter continues to be used for data entry and editing for the time being.

Revision to PSP re-measurement standards

A number of changes are recommended to PSP re-measurement standards, which are listed on pages 13-15. Because these require revision of forms and a field training exercise, it will not be possible to implement them until a new project is approved. Existing standards should continue to be used in the meantime. The new standards will eliminate inefficient extraneous measurements, concentrate effort on key variables, and improve site quality, topographic and geographic information.

Training course in Access

A 2-day training course in the use of Access was held for 11 participants, including three from PNGFA headquarters and eight from PNGFRI. The course contents are described in this report. As Access is available on almost every computer at PNGFRI, and data entry and storage is a problem for many studies, it was felt that this training would help a number of staff to begin using the facilities available to them.

The course was successful within the scope of the limited time available. The consultant would have preferred to have longer to prepare materials and handouts for the participants but this was not possible on this occasion to a fully satisfactory degree. For future reference the consultant suggests that the ratio of preparation to teaching time needs to be about 2:1, *ie.* two days preparation for each day of teaching. In this case, two days were spent preparing the two-day course.

General conclusions

The main output from this assignment has been the draft proposal for a new growth and yield project. That has been fully and satisfactorily completed. The draft is attached as a separate document. The remaining tasks undertaken were something of a compromise with the time available, but nonetheless should prove directly useful and represent an important preparation for the next phase of sustainable forest management studies in the natural forests of Papua New Guinea.

References

Alder, D; Oavika, F; Yosi, C (1998). Data, programs, and models for natural forest growth and yield. ITTO project PD 162/91 Internal Report, December 1998, 38 pp.

Romijn, K (1994a) PSP standards and procedures, parts A-D. ITTO Project PD 162/91 Internal Report.

Romijn, K (1994b) PERSYST: A data management system for permanent sample plots in natural forest. ITTO Project PD 162/91 Internal Report.